

第八节 持续集成和持续交付

大型互联网公司质量保证基本模式

项目开发工作流程平台

新建应用（基本信息） -> 申请代码权限（Git、SVN） -> 初始化代码 -> 提及集成 -> 申请测试环境 -> 验收测试环境 -> 发布预发布环境 -> 预发布验收 -> 线上做灰度测试（发布） -> 全面发布

发布频率，一个应用每周至少发布一次

带病上线，小步快跑，拥抱变化

代码托管

- SVN
- Git

- GitLab (基于开源实现, Ruby)
- 项目 Rebase
- 开发模式
 - 分支开发 (Master 稳定)
 - 主干为主, 非稳定
- Code Review
 - for-each lambda
- CI
 - GitLab CI

云效平台

- 代码编辑器
- 代码管理
- Code Review

单元测试

类的测试覆盖度 (9成)

- `Assert.assert(1 == 1);`

行的测试覆盖度

工具

- Junit 4 - 最常用
- Junit 5 - 支持并发测试（多次重复测试）
- TestNG - 支持并发测试
- Mock 工具
 - Mockito - 面向接口 Mock - 动态代理
 - 字节码提升

测试驱动开发

- 特性
 - 测试 -> 测试用例 -> 业务逻辑稳定
- 现实中的互联网
 - 业务不稳定

业务评审 -> 业务需求文档确定 -> 测试用例（描述性） -> 开发（测试用例实现）

持续集成工具

流水线 -> 工作流程 -> 阶段性运行

Pipeline -> Valve (阀门)

自动化

- 事件触发 (主动)
- 周期性触发
- 变化性触发 (被动)

工具推荐

- TravisCI
 - 通过 YML 配置初始化环境，构建虚拟资源 (Docker 镜像)，执行任务
- CircleCI
 -

Jenkins

运行方式

可执行 WAR

```
java -jar jenkins.war
```

初始化设置

<https://www.jenkins.io/doc/book/installing/initial-setting-s/>

插件安装和管理

JDK 安装

在 Global Configuration Tool

Maven 安装

创建工程

流行架构

Serverless 架构

- Web IDE
- 特别依赖于 CI/CD
 - Jenkins 被包装
 - 内核
 - Web GUI
- JMeter

国内外的项目管理差异

- 国外
 - 人力成本高（工具）
- 国内
 - 人工发布 -> 半自动化 -> 自动化（aone）
 - 智能运维
 - AI 运维

个人开发偏好

- 任何事件都是独立的，即使是靠谱的人，也有不靠谱的时候
- 功能开发都要确保单元测试和集成测试（可选）

- 单元测试类的覆盖度 $\geq 90\%$ ，行覆盖度 $\geq 80\%$
- 文档不能太少，基本要与代码持平（尤其是帮助文档）
- 用户输入是不可靠的，所以需要验证所有的条件分支

项目经历

TAE (Taobao Application Engine) , 参考 GAE, K8s

- DevOps
 - Code Base (Git、SVN)
 - GUI 管理界面
 - Code workspace
 - 自动化 (CI/CD)
- Tools
 - TAE SDK
 - 语言支持
 - PHP - Cauous
 - Java
 - Groovy
 - Scala
 - Ruby - JRuby
 - Python - Jython
 - JavaScript

推荐资料

书籍推荐

- 企业整合模式
- 敏捷开发
- XP
- 迭代开发 (Integration)
- 人月神话
- TDD (测试驱动开发)

互联网开发 - 简单粗暴，快速上线

SOFA - 基于 Spring Framework + OSGI

1.xml (a.jar)

```
<bean id="interceptor" class="..."></bean>
```

2.xml (b.jar)


```
<bean id="container">
    <property name="list">
        <bean id="interceptor1" class="...">
    </bean>
    </property>
</bean>
```

本次作业

1. 完善 my dependency-injection 模块
 - 脱离 web.xml 配置实现 ComponentContext 自动初始化
 - 使用独立模块并且能够在 user-web 中运行成功
2. 完善 my-configuration 模块
 - Config 对象如何能被 my-web-mvc 使用
 - 可能在 ServletContext 获取
 - 如何通过 ThreadLocal 获取
3. 去提前阅读 Servlet 规范中 Security 章节 (Servlet 容器安全)